ECE 445

SENIOR DESIGN LABORATORY

PROJECT PROPOSAL

Visual Chatting and Real-time Acting Robot

HAOZHE CHI
(haozhe4@illinois.edu)
ZONGHAI JING
(zonghai2@illinois.edu)
MINGHUA YANG
(minghua3@illinois.edu)
JIATONG LI
(jl180@illinois.edu)

March 7, 2024

1 Introduction

1.1 Problem Statement

The advent of Large Visual Language Models (LVLMs) marks a significant milestone in AI development, showcasing impressive capabilities in processing and understanding complex visual and textual data. Despite these advances, integrating LVLMs into robotics to enable seamless hardware coordination and direct action based on LVLM instructions remains a substantial challenge. This difficulty arises primarily because LVLM technology is relatively new, with comprehensive application methodologies, especially in robotics, still under exploration. Our project seeks to pioneer in this space by developing a robot that can interpret and act upon both audio and visual inputs through LVLM.

1.2 Solution Overview

Our goal is to create a robot capable of understanding human voice commands and visually interpreting its environment to perform tasks and respond verbally. The system architecture includes:

- A camera for real-time visual input.
- A speech-to-text AI model to convert voice commands into text.
- The BLIP-2 visual language AI model to process combined text and visual inputs, generating actionable instructions for robot movement and verbal responses.
- A text-to-speech AI model for converting textual responses back into audible speech, facilitated by a voice player for output.
- The Universal Robot Arm UR3e, chosen for its precision and flexibility, controlled via the Robot Operating System (ROS) based on LVLM instructions, with a laser module for enhanced object location accuracy.

1.3 Visual Aid

The overall networkk of our Robot-Computer System is shown in Figure 1.

1.4 High-level requirements list

1. A server equipped with a GPU of at least 24 GB of memory to efficiently run the visual language AI model.
2. A robotic arm with a minimum of six joints, specifically the Universal Robot UR3e, for versatile movement and object manipulation.
3. A ROS-compatible camera, like the CK camera, for capturing real-time visual data.

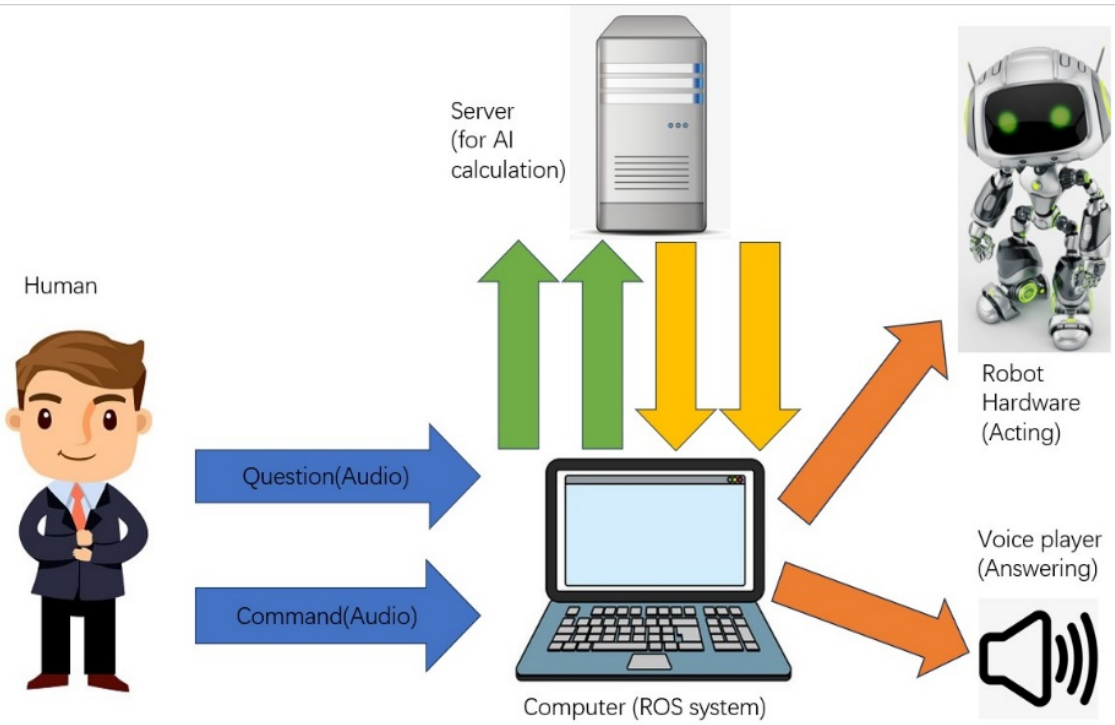

Figure 1: Overall Network of the Robot-Computer System

2 Design

2.1 Block Diagram

The overall block diagram of our Robot-Computer System is shown in Figure 2.

2.2 Subsystem Overview

Q-Former Subsystem The Q-Former subsystem, based on transformer architecture, transforms visual features into more abstract representations by incorporating attention mechanisms. It receives visual features from the Image Encoder subsystem and relays enhanced features to the Large Language Model subsystem for further processing.

Image Encoder Subsystem Utilizing a Vision Transformer (ViT) structure, this subsystem encodes images into visual features for analysis. It captures images from the Camera subsystem and forwards the processed visual features to the Q-Former subsystem.

Text Tokenizer Subsystem This subsystem tokenizes and converts input text into embeddings, facilitating textual analysis. Although it is mentioned that it receives images from the Camera subsystem, it should correctly receive text inputs, likely from the Speech-to-Text subsystem, and send the resulting embeddings to the Large Language Model subsystem.

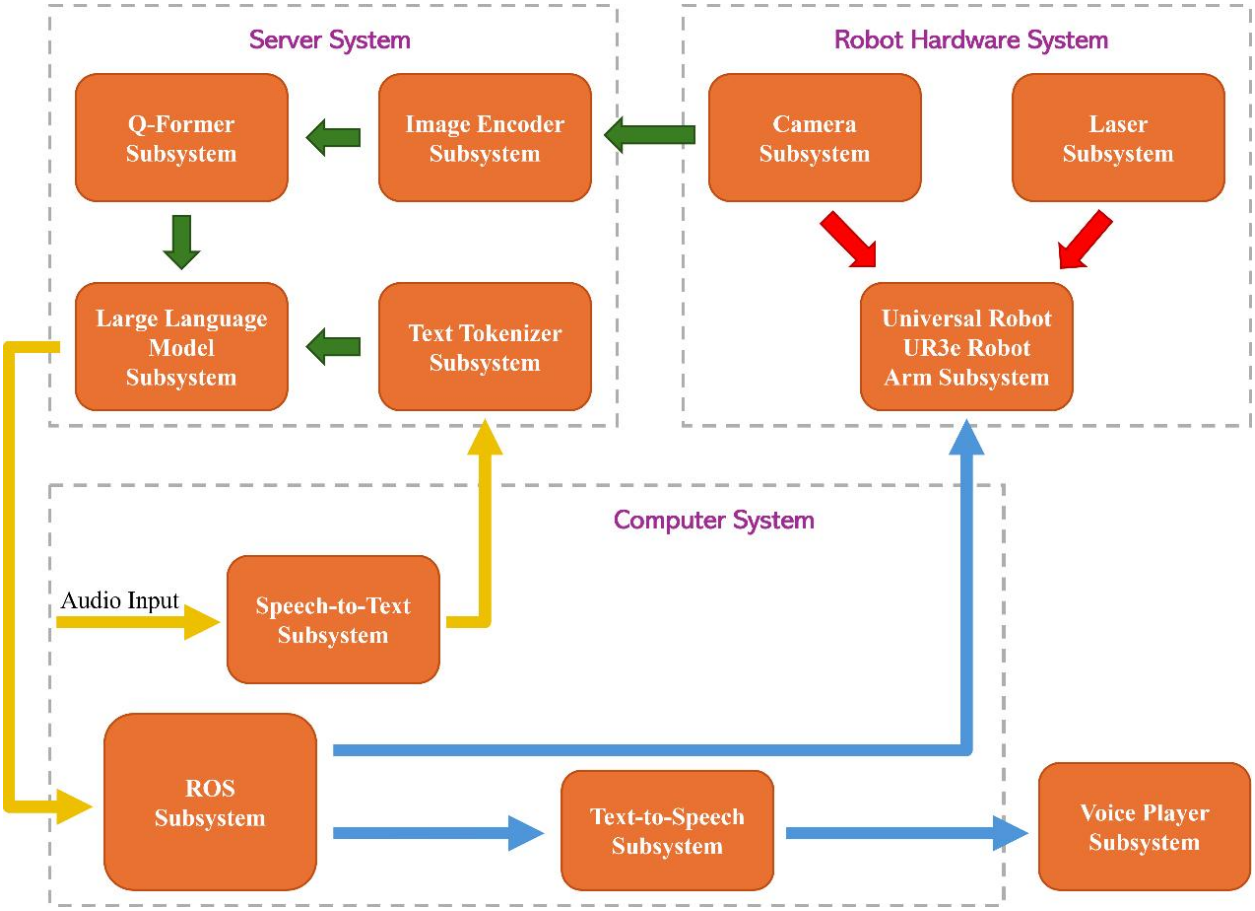

Figure 2: Overall Block Diagram of the Robot System (Green arrow: visual flow; Yellow arrow: text flow; Red arrow: attach itself to; Blue arrow: instruction flow)

Large Language Model Subsystem Processing both visual and textual embeddings, the Large Language Model subsystem generates text outputs that encompass instructions for robotic actions and responses to queries. These outputs are directed to the ROS subsystem for action and response articulation.

Camera Subsystem Operating continuously, the Camera subsystem captures images every second, providing real-time visual data to the Image Encoder subsystem for processing.

Laser Subsystem Integrated with the Universal Robot UR3e robot arm, the Laser subsystem enhances the robot's precision in object localization, facilitating accurate interaction with its environment.

Universal Robot UR3e Robot Arm Subsystem Receiving instructions from the ROS subsystem, the UR3e robot arm interacts with its surroundings, leveraging data from the Camera and Laser subsystems to move and manipulate objects as directed.

Speech-to-Text Subsystem This subsystem converts spoken language into textual format, supplying the Text Tokenizer subsystem with text for embedding and further processing.

ROS Subsystem As the control hub, the ROS subsystem manages hardware operations based on instructions and responses generated by the Large Language Model subsystem. It orchestrates the execution of robotic actions and relays responses to the Text-to-Speech subsystem.

Text-to-Speech Subsystem Converting textual responses into audible speech, this subsystem ensures the robot can verbally communicate with users, passing audio outputs to the Voice Player subsystem for playback.

Voice Player Subsystem This subsystem is responsible for audibly playing back responses generated by the Text-to-Speech subsystem, enabling interactive communication between the robot and its human users.

2.3 Subsystem Requirements

Q-Former Subsystem The Q-Former subsystem acts as an integrative bridge, merging the capabilities of a static image encoder with a static Large Language Model (LLM). It is designed to extract and refine output features from the image encoder, irrespective of the input image's resolution. This subsystem comprises two transformer submodules utilizing shared self-attention layers, facilitating efficient processing of visual features and textual data. The image transformer submodule is dedicated to processing visual information received from the image encoder, while the text transformer submodule functions dually as an encoder and a decoder for textual information. This dual functionality enables the Q-Former to seamlessly integrate visual and textual inputs for comprehensive processing.

Image Encoder Subsystem The Image Encoder subsystem employs a transformer-based architecture, specifically a **ViT-L/14** model, to encode images into a compact, feature-rich representation. It utilizes 32 distinct queries, each with a dimension of 768, matching the Q-Former's hidden dimension. The output, denoted as Z, adopts a 32×768 dimension, significantly reducing the representation size compared to the original image features extracted by the ViT-L/14 (257×1024), thus enhancing processing efficiency.

Text Tokenizer Subsystem The Text Tokenizer subsystem is responsible for converting raw text input into structured embeddings, using a tokenizer consistent with the **BERT** model. This approach ensures compatibility with widely utilized Large Language Models, such as Llama, facilitating seamless integration within the broader architecture and promoting effective textual analysis.

Large Language Model Subsystem This subsystem leverages an open-source Large Language Model, such as Llama, serving as a downstream decoder to process the combined input features from both visual and textual sources. The choice of **Llama** aligns with the architectural principles of BLIP-2 [1] based systems, enabling sophisticated text output generation that includes both instructions for robotic actions and responses to user inquiries.

Camera Subsystem The camera subsystem incorporates a CK camera, selected for its compatibility with the ROS ecosystem. It is tasked with capturing real-time images, ensuring the robot can react to its environment with updated visual information.

Laser Subsystem The Laser subsystem is designed to augment the robot arm's ability to accurately locate and interact with objects. Attached to the Universal Robot UR3e arm, it enhances precision in object manipulation tasks.

Universal Robot UR3e Robot Arm Subsystem Utilizing the Universal Robot UR3e, this subsystem is central to the robot's physical interactions with its environment. The UR3e is chosen for its flexibility and precision, featuring six joints that facilitate a wide range of movements and tasks.

Speech-to-Text Subsystem This subsystem employs an open-source model, such as those available from Google, to convert human speech into text. The processed textual data is then relayed to the Text Tokenizer subsystem, ensuring that voice commands are accurately interpreted and acted upon.

ROS Subsystem The Robot Operating System (ROS) subsystem serves as the robot's meta-operating system, offering essential services including hardware abstraction, device control, and inter-process communication. ROS's comprehensive ecosystem supports the development and execution of robotic applications across diverse hardware setups.

Text-to-Speech Subsystem Employing ***pyttsx3***[1], an open-source Python library for offline text-to-speech conversion, this subsystem transforms text outputs from the ROS subsystem into audible speech. This allows the robot to communicate responses to user queries verbally, enhancing the interactive experience.

Voice Player Subsystem The Voice player subsystem is responsible for the audible output of the robot's responses. It receives audio files from the Text-to-speech subsystem and plays them out loud, enabling the robot to communicate effectively with its human users.

[1]https://pyttsx3.readthedocs.io/en/latest/

2.4 Tolerance Analysis

A key design consideration is the latency involved in data transfer between the robot and the server, which is critical for real-time interaction and control. Through simulations, we have identified that data transfer delays can be minimized to approximately 10 milliseconds. Additionally, employing Python libraries such as flash-attention can significantly enhance the processing speed of our AI models. Experimental results suggest that the overall processing time, though variable, remains within a few seconds based on the complexity of input data, which is within acceptable limits for real-time operation.

3 Ethics and Safety

As we work on our robot project, we're keeping a close eye on doing things in the right way according to the IEEE Code of Ethics [2].

3.1 Ethics

Privacy (ACM 1.7: Respect the privacy of others) We're making sure that any personal information the robot picks up is kept safe and private. Only people who really need to see this info can access it, and we're careful about how we store and handle it.

Fairness (IEEE - Avoiding Real or Perceived Conflicts of Interest) Sometimes, AI can be biased, making unfair decisions. We're working hard to train our robot with a wide variety of data so it treats everyone fairly.

Being Open (ACM 1.2: Avoid harm) We want everyone to understand how our robot makes decisions. We're keeping records and will be open about how the robot works and what it's been taught.

3.2 Safety

Avoiding Accidents (IEEE - Priority to Public Welfare) Our robot is built to be safe around people and objects, with emergency stops and sensors to keep it from bumping into things.

Staying Secure (ACM 3.7: Recognize the need to protect personal data) Since our robot is smart and connected, we're putting in strong security to protect it from hackers.

Dealing with Mistakes (ACM 2.5 & IEEE - Acknowledge and Correct Mistakes) If something goes wrong, the robot is designed to handle it safely and let us know so we can fix the problem.

References

[1] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," *arXiv preprint arXiv:2301.12597*, 2023.

[2] Institute of Electrical and Electronics Engineers. (2016) IEEE Code of Ethics. Accessed on: 2024-03-07. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html